Serial Interface Description

Electromechanical locking device HD-Lock - Art. No. 102000.2

Series (CAN Bus/Modbus)

Table of contents

Operating instructions Locking device HD-Lock

Item no.: 102000.2
Version: 1.1
Date: 15th August 2022
Document: 102000SerSch_ENG

1 General

The HD-Locks of the 102000.1 and 102000.2 series have a serial interface in the form of a differential wire pair. Both series can communicate via this interface using the "Remote Terminal Unit Modbus®" protocol (RTU-Modbus®). In addition, the serial interface of the 102000.2 series can be switched to the CAN bus protocol. Further information can be found in the chapter 5.

2 Memory structure

With HD-Lock, the memory is divided into two different register types. Each type has its own address space and different read/write permissions, which is described below:

- **Input register:** 16-bit registers that can only be read. Sensor values and system parameters are stored in this memory area. These are mainly used for troubleshooting and manual function testing.
- **Holding register:** 16-bit registers that can be read and written. The registers in this area are used for parameterisation and thus for influencing the firmware. However, most of the registers in this area are protected against write access by the firmware of the HD-Lock in order to prevent an unintentional change of the parameters and to ensure proper functioning. Any changes to the holding registers are only stored volatile. To achieve non-volatile storage, it is necessary to send the "Save Settings" system command. For more information, see section 3 Operation.

3 Operation

The following shows the operating options that are available for the HD-Lock.

3.1 System commands

The basic control commands of the HD-Lock are summarised under "System commands". A system command includes an ID number (see below) that identifies the command and up to two transfer parameters that are processed by the command. The holding registers 2000 to 2003 are used to enter these commands. For correct execution of a command, if necessary, the transfer parameters must first be set (holding registers 2000 to 2002) before the identification number for the command is sent (per holding register 2003). The following is a list of the commands:

Command	ID number (Holding Register 2003)	Parameter 0 (Holding Register 2000)	Parameter 1 (Holding Register 2001)	Parameter 2 (Holding Register 2002)
Save Settings	1			
Restore Settings	2			
Set Modbus Address	3	New address	Condition 1 (register 1000)	Condition 2 (register 1000)

Table 1: System commands

Save Settings

This command takes over the contents of the holding registers into the non-volatile memory so that they are reloaded after a reset.

Restore Settings

With the "Restore Settings" command, the non-volatile stored values of the holding registers are reloaded. All changes that have not yet been transferred to the non-volatile memory are lost.

Set Modbus Address

The "Set Modbus Address" command can be used to change the Modbus address of the HD-Lock. First, the three parameters must be entered:

- Parameter 0 must contain the new address. (Only addresses from 1 to 254 are allowed).
- Parameter 1 corresponds to the selection bit mask which selects the conditions to be evaluated (holding register 1000, see chapter 4.3 selects the conditions.
- Parameter 2 corresponds to the evaluation bit mask that determines the truth value of the previously selected conditions.

The condition(s) entered by parameter 1 is/are linked to the current status of the HD-Lock bitwise AND. The result of this link is then linked with parameter 2 Bitwise-Exclusive-ODER (XOR). If this results in a value equal to zero, the new address is taken over. This operation is described in short form below:

$$(Parameter\ 1\ \wedge StatusHdLock) \underline{\vee}\ Parameter\ 2 = 0$$

Formula 1: Matching conditions

$[\wedge =\ \text{Bitweise} - \text{UND};\ \underline{\vee} = \text{Bitweise} - \text{XOR}]$

Only if these conditions result in 0, the address is changed.

Several conditions from the status register can also be combined in parameters 1 and 2.

For example, the following conditions can be selected via parameter 1 = 0x0118:

- Bolt fully extended?
- Is there a locking request?
- Are the Hall sensors being overridden?

Parameter 2 then determines whether the conditions are to be fulfilled or not.

If the condition "bolt fully extended?" was selected with parameter 1 = 0x0008 and parameter 2 = 0x0100, the bolt must be fully extended for the address to be accepted.

However, if parameter 2 is equal to 0x0100, the bolt must <u>not</u> be fully extended if the address is to be adopted.

Example for taking over the new address:

Parameter 0 = 0x0008 (New address)

Parameter 1 = 0x0108 (Selection: "Bolt extended?", "Hall sensors overridden?")

Parameter 2 = 0x0100 (status word "bolt extended", "Hall sensors <u>not</u> overridden")

Command = 3

Now the software checks whether the bolt of the HD-Lock is extended (parameter 1 & StatusHdLock). This intermediate result is now compared with parameter 2 (bitwise XOR). The final result must be a value equal to 0, then the address 0x0008 in the holding register 103 becomes effective.

To save it permanently, the "Save Settings" command must be sent to the new address.

	0x0108 (0000 0001 0000 1000)	Parameter 1 (selection bit mask)
^	**<u>0x8117</u>** <u>(1000 0001 0001 0111)</u>	Status HD-Lock (structure status word see chapter 5.1)
	0x0100 (0000 0001 0000 0000)	Intermediate result
⊻	**<u>0x0100</u>** <u>(0000 0001 0000 0000)</u>	Parameter 2 (evaluation bit mask)
	0x0000 (0000 0000 0000 0000)	Final result (equal to 0 the address is taken over)

This makes it possible, for example, to address all HD locks on the bus at the same time via a broadcast telegram to address 255 and only the one for which the condition is set according to the comparison values takes over the new address.

3.2 Output functions

The outputs of the HD-Lock can be used to display various information. For this purpose, output functions assign a register value to a display. The factory setting for output 1 is the function "DoorLocked" (register value = 4) and for output 2 the function "DoorClosed" (register value = 2).

The following assignments are possible:
Register value 0→ Always off
Register value 1→ Always on
Register value 2→ On when door is closed
Register value 3→ On, if door is **not** closed
Register value 4→ On when door is locked
Register value 5→ On when door is **not** locked
Register value 6→ On, if Locking request (input "Effective lock request") is present
Register value 7→ On, if **no** lock request (input "Effective lock request") is available
Register value8→ on/off periodically (approx. 1 Hz).
Register value9→ reserved
Register value 10→ On, if Hall sensors are overridden.

Remark: On = Output has voltage level of the supply voltage
Off = Output has voltage level of ground.

To assign a new display to an output, the function values for output 1 [holding register 3001] and output 2 [holding register 3002] must be entered. To save the setting permanently, the "Save Settings" command must be sent.

3.3 Direction Function

"Direction" is used to switch the direction of impact of the HD-Lock (see Figure 1). The **automatic detection of the direction of impact is activated in the factory.** This has the value "Direction" = 0 (holding register 3003). If the direction of impact is to be fixed, this value can be changed. In the following illustration the values to be set are explained on the basis of the direction of impact (looking at the mirror-polished surface of the locking device in the installed state):

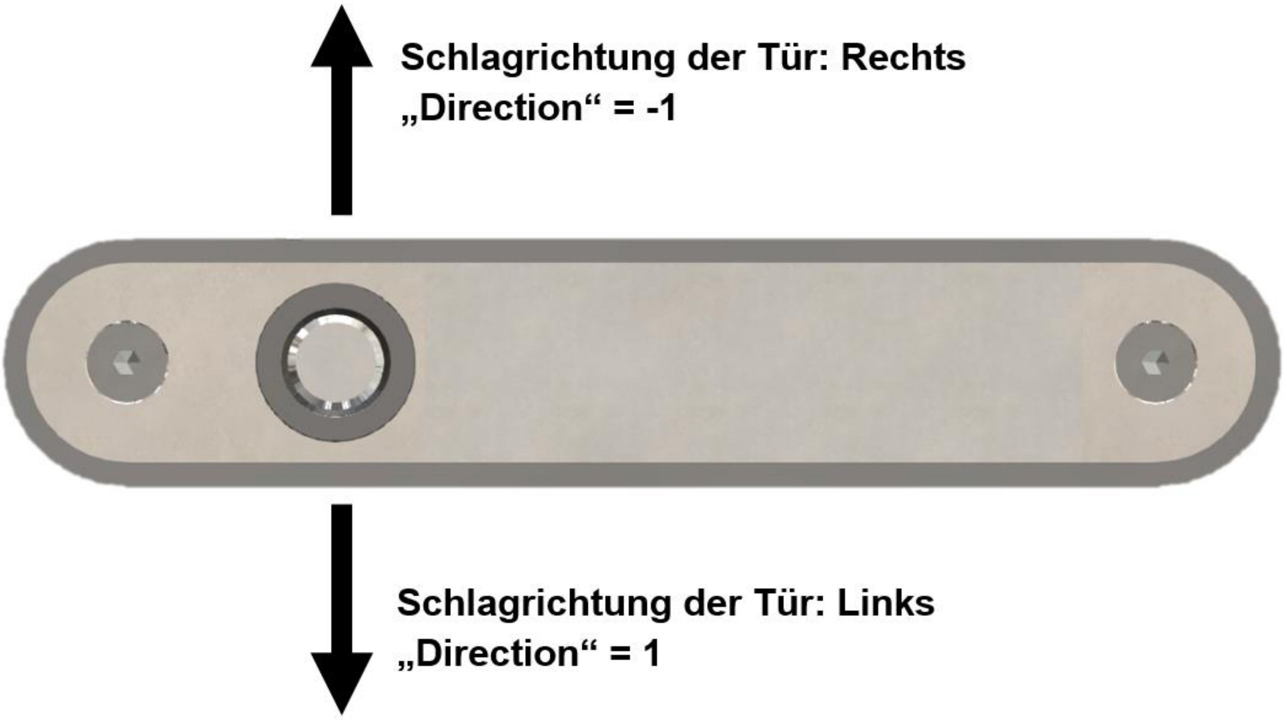

Figure 1 : Setting the "Direction-Function" depending on the direction of the door's stroke

<u>IMPORTANT!</u> It is imperative that the direction of impact is set correctly, otherwise the locking device may malfunction. After restarting the HD-Lock, the door must be opened once and closed again so that the automatic detection can be carried out.

3.4 Proximity Threshold

The following registers can be used to set the switching points for the open/close detection of the door:

- Holding register 3004 "SettingsProximityThresholdUnlock" (factory setting = -30)
- Holding register 3006 "SettingsProximityThresholdLock" (factory setting = -60)

The Proximity Threshold value and thus the switching point of the open/close detection of the door is differentiated in two states:

- Door unlocked" status (bolt retracted [holding register 3004])

  When the door is closed, the locking device detects the striking plate. To ensure precise extension of the bolt, the bolt is only extended when the striking plate is almost congruent with the locking device. When the door strikes from the other side, the distance is mirrored to the other side (see Figure 2).

Figure 2 : Coinciding distance between locking device and striking plate

- Door locked" status (bolt extended [holding register 3006])

  If the door is locked, the door detection window is enlarged. This means that the door is recognised as "closed" even if the handle is operated and the door is shaken. If the extended window for detection is left, it can be assumed that the door has been opened by force.

Due to installation tolerances, it may happen that the factory-set switching point is not reached. In this case, the value "SettingsProximityThresholdUnlock" and "SettingsProximityThresholdLock" must be corrected to ensure error-free operation.

The following illustration can be used for this setting.

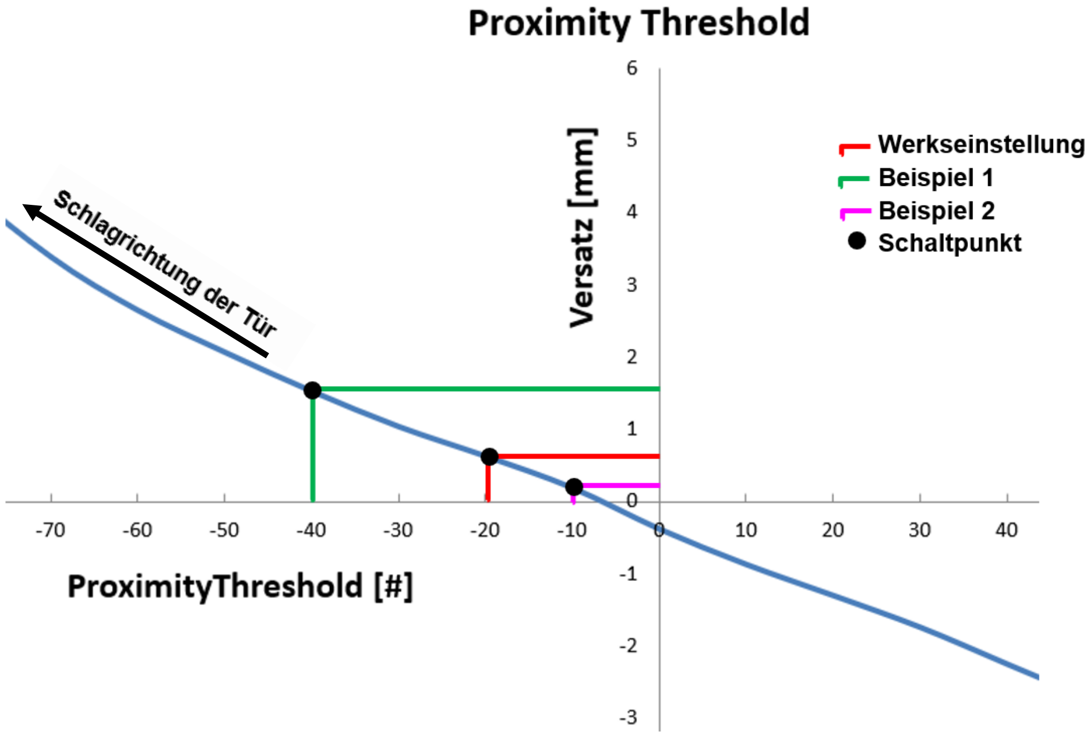

Figure 3 : Effect of the "Proximity Threshold" value on the offset to the striking plate.

It should be noted that the detection of the striking plate position depends on the distance between the striking plate and the HD-Lock. The further away the striking plate is from the HD-Lock, the greater the offset in millimetres with a constant "proximity" value. The curve Figure 3 curve shown was determined at a distance of 8.3 mm,

In addition, a highly accurate position detection can be set via offset values. This is only necessary in special cases and can be set via the holding registers 3007 and 3008. The following illustration shows this adjustment. The green arrows apply to a door opening on the right and the yellow arrows apply to a door opening on the left. A positive value in one of the registers moves the zero line along the arrows marked with "+". A negative value moves it along the arrows marked with "-".

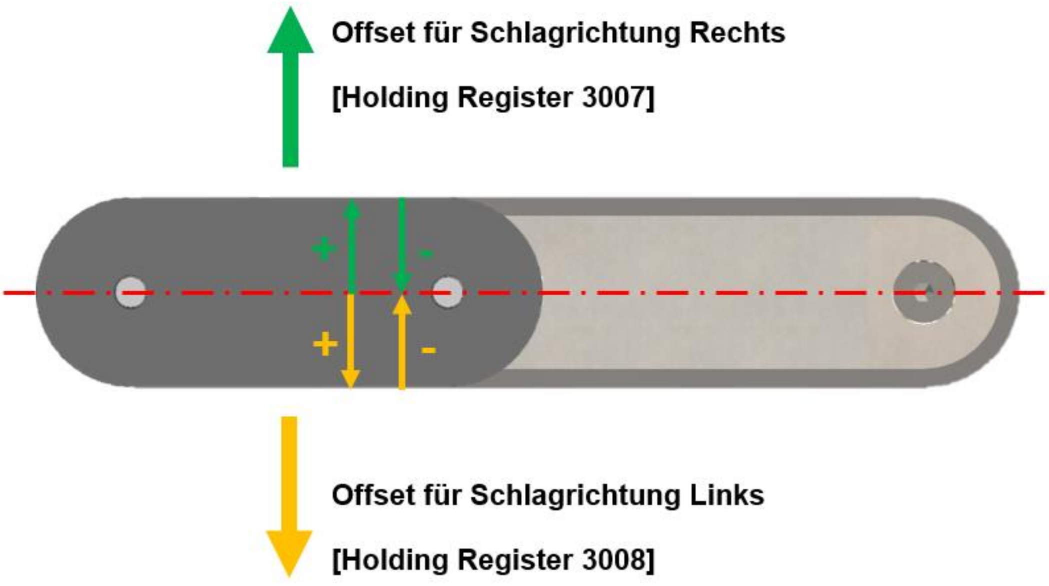

Figure 4: Offset setting of congruency

3.5 Remove Magnet Feature

By changing this parameter, you can control whether the stainless steel bolt is retracted as soon as the magnetic field of the strike plate is weakened. The factory setting for this parameter is "0" (holding register 3005), i.e. if the bolt is extended, the magnetic field can be weakened and the bolt will not retract. If this function is desired, the value can be set to "1" and the bolt will retract when the magnetic field weakens, despite the locking request.

ATTENTION If this function is set to "1", there is a safety risk as the locking device can be unlocked by manipulation from the outside!

3.6 Lock Reqest Delay

If the striking plate is detected by the sensors and the locking position is reached, a time delay is activated to allow for possible door swinging. This time delay can be changed with the value "LockRequestDelay" (holding register 3000). The factory setting for this value is 200 [ms].

3.7 Lock Request Source

The behaviour of the door lock can be controlled by entering a specific value in the holding register 1002 "Lock Request Source". The following table lists the conditions that must be fulfilled for the bolt to extend. A distinction is made here according to whether the door is closed (strike plate with magnet in position) or not:

Value	**Designation**	**Condition 1**	**Condition 2**
0x00	IrsNever	Never fulfilled	Door closed
0x01	IrsAlways	Always fulfilled	Door closed
0x02	IrsOverrange	Met if Hall sensors are overridden (bit 3 of input register 1000 active).	Door closed
0x03	IrsLine	Met if input **inactive [<4V]** (bit 0 of input register 1000)	Door closed
0x04	IrsNotLine	Met if input **active [>10V]** (bit 0 of input register 1000)	Door closed
0x05	IrsToggleOnLineEdge	Met if an edge is detected at the input (bit 0 of input register 1000).	Door closed
0x06	IrsToggleOnLinePEdge	Met if a positive edge is detected at the input (bit 0 of input register 1000).	Door closed
0x07	IrsToggleOnLineNEdge	Met if a negative edge is detected at the input (bit 0 of input register 1000).	Door closed
0x80	IrsNever	Never fulfilled	None
0x81	IrsAlways	Always fulfilled	None
0x82	IrsOverrange	Met if Hall sensors are overridden (bit 3 of input register 1000 active).	None
0x83	IrsLine	Met if input **inactive [<4V]** (bit 0 of Input- Register 1000)	None
0x84	IrsNotLine	Met if input **active [>10V]** (bit 0 of Input- Register 1000)	None
0x85	IrsToggleOnLineEdge	Met if an edge is detected at the input (bit 0 of input register 1000).	None
0x86	IrsToggleOnLinePEdge	Met if a positive edge is detected at the input (bit 0 of input register 1000).	None
0x87	IrsToggleOnLineNEdge	Met if a negative edge is detected at the input (bit 0 of input register 1000).	None

Table 2: Values for "Lock Request Source

4 Register map

The different registers are listed below. These are divided into input registers and holding registers.

4.1 Input register

(16-bit registers that can only be read)

Address	***Name***	***Description***
0	LockState	Indicates the current locking status 0 = open (bolt retracted) 1 = closed (bolt extended)
1	LockRequest	Returns the current interlock request 0 = Interlock request inactive 1 = Interlock request active
2	BoltPosition	Indicates the current position of the bolt (increments) 0 (retracted) 135 (max. value in extended state)
3	MotorState	Indicates the current travel direction of the motor 0 = inactive 1 = extending 2 = retracting 3 = braking
4	MotorSpeed	Indicates the current travel speed of the motor
5	ValueHallSensor1	Indicates the current value of Hall sensor 1
6	ValueHallSensor2	Indicates the current value of Hall sensor 2
7	Calculated proxi (normalised)	Calculated position of the magnet via the values of Hall sensor 1 and Hall sensor 2 (see chapter 5.5 Proximity Threshold).
8	DoorState	Indicates the current status of the door 0 = undefined 1 = door open 2 = door close 3 = door closed
9	StateMachineState	Indicates the current status of the state machine of the locking

		device: 0 = start 1 = referencing 2 = opened 3 = locking 4 = locked 5 = open 6 = error locked 7 = error locking 8 = error locking pause short 9 = error locking pause long 10 = error locking wait 11 = error opened 12 = error open
10	StateMachineElapsed	Indicates the elapsed time since the last change of the "StateMachineState" value [Input Register 9].
11	Direction	Indicates the current stroke direction: -1 = always right impact 0 = automatic mode 1 = always left impact See chapter 3.4 Proximity Threshold
12	RetryLockCounter	Indicates the number of locking attempts in which the bolt cannot extend properly. Locking attempts are set via holding registers 32 and 33.
13	RetryUnlockCounter	Indicates the number of unlocking attempts during which the bolt cannot retract properly. Unlocking attempts are set via holding registers 32 and 33.
14	SystemCoreClock[MHz]	Indicates the current frequency of the microcontroller in MHz
15	SystemCoreClock-Config	Indicates the current frequency source: Bit 0: HSI ready (internal frequency encoder ready) Bit 1: HSE ready (external frequency encoder ready) Bit 2: PLL ready (phase-locked loop ready) Bit 3: 0 = HSI (internal frequency encoder active) 1 = HSE (external frequency generator active) Bit 4-5: System clock source 0 = HSI (internal frequency encoder is used) 1 = HSE (external frequency generator is used) 2 = PLL (phase-locked loop is used)
16	Selected Threshold	Displays the Proximity Threshold currently in use. This must correspond to either the "SettingsProximityThresholdUnlock" value [Holding Register 3004] or the "SettingsProximityThresholdLock" value [Holding Register 3006].
17-49	Reserved	
50	ActS	Current motor position [#]
51	ActV	Current motor speed [#/s]

52	MeasuredV	Motor speed measured [#/s]
53	Motor State	Current engine condition -1 = Inactive 0 = Forward 1 = Backwards 2 = Braked 3 = Last state 4 = Active
54	Motor Result	Reason for last engine stop None AtPos MinSpeed Timeout userStop OutStageDisabled Currentlimit
55	Act PWM	Currently active PWM [%]
56	Voltage at Motorclamps	Current voltage at the motor (VCC x PWM) [V].
57	actCurrent	Current motor current [mA]
58	maxCurrent	Highest detected motor current [mA]
59	errVsum	Integral part of the speed controller
60	Time since last change	Elapsed time since the engine status last changed.
61	Last time since last change	Previous value of "Time since last change" [Holding register 60].
62	Supply Voltage	Indicates the current supply voltage
63	Act current	Indicates the current motor current
64-99	Reserved	
100	Firmware version	Indicates the current firmware version, which is described in the decimal system, in 16 bits.
101	SavedSettings	Specifies the number of times the memory has been overwritten. As soon as the Save Setting command is executed, this value is incremented.
102	BootloaderVersion	Indicates the current bootloader version.
103	HardwareVersion	Indicates the current hardware version.
104	SerialNumber	Indicates the serial number/device ID of the unit.

Table 3: Register map input register

4.2 Holding register

(16-bit registers that can generally be read and written (RW). To protect system-relevant registers from unintentional modification, the firmware implements an access restriction. This means that several registers can only be read (RO). The access restriction can be temporarily lifted by trained BSS personnel, whereby registers with the note "manufacturer" can also be written to.

Changes must be transferred to the non-volatile memory with the Save-Settings command!

Address	*Name*	*Description*	*Access*
0	SettingsRevision	Change the memory setting. Indicates the number of times memory has been saved. This value can be changed as follows. Decrement 1 = Factory setting 0 = Load from flash memory Increment 1 = Store in flash memory	RO / Manufacturer
1	LockRequest	Recognition of the inputs 0= low Active 1= high Active 2 = Toggle	RO / Manufacturer
2	BoltPosition	Indicates the current position of the motor (increments) 0 (retracted) 135 (max. value in extended state)	
3	ZeroPositionWhenUnlocked	Indicates whether the bolt position is automatically zeroed when the bolt is unlocked. 0 =	
4	LockRequestSource	Selection to control the locking (see chapter 3.7 Lock Request Source)	
5-9	Reserved		
10	LockTargetPosition	Locking position value that should be reached by the bolt	RO / Manufacturer
11	LockOkPosition	Interlock position value from which an interlock is established	RO / Manufacturer
12	LockTargetSpeed	Traversing speed to be reached during locking	RO / Manufacturer
13	LockAcceleration	Acceleration when extending the bolt from standstill to traversing speed	RO / Manufacturer
14	LockDeceleration	Braking deceleration when extending the bolt shortly before reaching the end position	RO / Manufacturer

15	LockHoldingPWM	PWM, with which the motor is controlled in order to hold the bolt in position.	RO / Manufacturer
16	LockMaximumTime	Locking time, measured from signal input to the time when the bolt has reached its end position [ms].	RO / Manufacturer
17	LockMaximumCurrent	Maximum current during locking [ADC#]	RO / Manufacturer
18	LockMinimumSpeed	Minimum speed when locking [#/s].	RO / Manufacturer
19	LockDisturbanceDelay	Interference delay of the locking process [ms]	RO / Manufacturer
20	UnlockTargetPosition	Unlock position value that should be reached by the bolt	RO / Manufacturer
21	UnlockOkPosition	Unlock position value from which an interlock is established	RO / Manufacturer
22	UnlockTargetSpeed	Traversing speed to be reached during unlocking	RO / Manufacturer
23	UnlockAcceleration	Acceleration when retracting the bolt from standstill to traversing speed	RO / Manufacturer
24	UnlockDeceleration	Braking deceleration when retracting the bolt shortly before reaching the end position	RO / Manufacturer
25	UnlockHoldingPWM	PWM, with which the motor is controlled in order to hold the bolt in position.	RO / Manufacturer
26	UnlockMaximumTime	Locking time, measured from signal input to the time when the bolt has reached its end position [ms].	RO / Manufacturer
27	UnlockMaximumCurrent	Maximum current during unlocking	RO / Manufacturer
28	UnlockMinimumSpeed	Minimum speed during unlocking	RO / Manufacturer
29	UnlockDisturbanceDelay	Interference delay of the locking process	RO / Manufacturer
30	Lock/UnlockFailDelay1	Time between locking and unlocking attempts [ms] (interval 1)	RW
31	Lock/UnlockFailDelay2	Time between repetitions of interval 1 [s] (interval 2)	RW
32	Lock/UnlockInterval1	Number of locking and unlocking attempts which are carried out with an interval of LockFailDelay1 [Holding register 30].	RW
33	Lock/UnlockInterval2	Number of repetitions of interval 1, which are executed with an interval of LockFailDelay2	RW

		[Holding register 31] .	
34-39	Reserved		
40	Output 1 Function	Displays the "Output function" selected for output 1. (Default: 4)	RO / Manufacturer
41	Output 2 Function	Displays the "Output function" selected for output 2. (Default: 2)	RO / Manufacturer
42-99	Reserved		
100	Baud rate	Setting of the Modbus baud rate. Valid values are: 300, 1200, 4800, 9600, 14400, 19200, 38400, 56000, 57600 (Default: 19200)	RO
101	Parity	Setting of the parity bit of the modbus. Valid values are: 0 = none, 1=odd, 2 = even (default: 2)	RO / Manufacturer
102	StopBit	Number of StopBits of the modbus. Valid values are: (0 = 0.5 bit, 1 = 1 bit, 2 = 2.5, 3 = 2	RO / Manufacturer
103	DeviceAddress	Setting of the unit address only possible via holding register 2000-2003. See chapter 3.1	RO / Manufacturer
104-999	Reserved		
1000	DoorState	Indicates the status of the door in 16-bit binary. See section 4.3, where the individual bits are explained.	RO
1001	BoltPosition	Indicates the current position of the bolt (increments) 0 (retracted) 135 (max. value in extended state)	RO
1002	LockRequestSource	Selection of the conditions under which the pin extends. (see chapter 3.7)	RW
1003	MaximumTimeUnlock	Longest time measured so far that the bolt needs to unlock [ms].	RW
1004	MaximumTimeLock	Longest time measured so far that the bolt needs to lock [ms].	RW
1005	ValueProximityThreshold	Calculated position of the magnet via the values of Hall sensor 1 and Hall sensor 2 (See chapter 3.4 Proximity Threshold)	RO
1006	Hall	Analogue values of the Hall sensors:	RO

<table>
<tr><td></td><td></td><td>$H1_{[7...0]}$: 8 bit analogue value of sensor 1
$H2_{[7...0]}$: 8 bit analogue value of sensor 2
Register content:
[$H1_7$, $H1_6$, $H1_5$, $H1_4$, $H1_3$, $H1_2$, $H1_1$, $H1_0$, $H2_7$, $H2_6$, $H2_5$, $H2_4$, $H2_3$, $H2_2$, $H2_1$, $H2_0$]</td><td></td></tr>
<tr><td>1007</td><td>Temperature</td><td>Temperature measured by the CPU [°C].</td><td>RO</td></tr>
<tr><td>1008</td><td>Voltage</td><td>Voltage value of the supply voltage [mV]</td><td>RO</td></tr>
<tr><td>1009</td><td>MotorCurrent</td><td>Current motor current as ADC value [mA]</td><td>RO</td></tr>
<tr><td>1010</td><td>MaximumMotorCurrent</td><td>Maximum motor current as ADC value [mA]</td><td>RO</td></tr>
<tr><td>1011-1999</td><td>Reserved</td><td></td><td></td></tr>
<tr><td>2000</td><td>Parameter0</td><td>Register for entering the parameter 0, which is required for some system commands. (Default: 0)<table><tr><td>Command</td><td>Possible value</td></tr><tr><td>Set Modbus Address</td><td>Modbus address to be reassigned in the range 1 - 254</td></tr></table></td><td>RW</td></tr>
<tr><td>2001</td><td>Parameter1</td><td>Register for entering parameter 1, which is required for some system commands. (Default: 0)<table><tr><td>Command</td><td>Possible value</td></tr><tr><td>Set Modbus Address</td><td>Input comparison value</td></tr></table></td><td>RW</td></tr>
<tr><td>2002</td><td>Parameter2</td><td>Register for entering parameter 2 (default: 0)</td><td>RW</td></tr>
<tr><td>2003</td><td>Command</td><td>Register for entering the command that is required for some system commands. (Default: 0)<table><tr><td>Command</td><td>Value</td></tr><tr><td>Save Settings</td><td>1</td></tr><tr><td>Restore Settings</td><td>2</td></tr><tr><td>Set Modbus Address</td><td>3</td></tr></table></td><td>RW</td></tr>
<tr><td>2004-2999</td><td>Reserved</td><td></td><td></td></tr>
<tr><td>3000</td><td>DoorClosedDelay</td><td>Delay of the bolt when extending after the door has been detected [ms].</td><td>RW</td></tr>
<tr><td>3001</td><td>SettingsOutput1</td><td>Setting of output 1 (PIN 1, wire colour brown) at which signal it switches to a high signal. Signals are described in chapter 3.2 described</td><td>RW</td></tr>
</table>

3002	SettingsOutput2	Setting of output 2 (PIN 3, wire colour white) at which signal it switches to a high signal. Signals are described in chapter 3.2 described	RW
3003	SettingsDirection	Setting the Direction function (see chapter 3.3)	RW
3004	SettingsProximityThreshold	Adjustment of congruence of striking plate and locking device see chapter 3.4	RW
3005	SettingsRemoveMagnet	Adjustment of the bolt retraction when the magnetic field changes: 0 = Bolt is not automatically retracted when the magnetic field changes, but is only controlled via interlocking request input register 1 1 = Bolt is automatically retracted when the magnetic field weakens (ATTENTION: safety risk)	RW
3007	SettingsProximityOffsetLeft	Setting the offset value from input register 11 "Direction" for doors opening to the left	RW
3008	SettingsProximityOffsetRight	Setting the offset value from input register 11 "Direction" for doors opening to the right	RW
3009-32769	Reserved		RW
32770	FwUpdateCommandRegister	Used to trigger a restart for a firmware update. 1 = Reset	RW

Table 4: Register Map Holding Registers

4.3 Status Word

Holding register 1000

(bit mask [hex])	*Name*	*Description*
0x0001	InputActive	Active when digital input active
0x0002	ProximityInPlace	Active when strike plate (door) in position
0x0004	DoorInPosition	Active when strike plate (door) in position with fault delay
0x0008	ProximityOverrange	Active when both Hall sensors are overridden (strong magnet on the sensors).
0x0010	LockRequest	Active when locking request active
0x0020	Reserved	-
0x0040	Reserved	-
0x0080	ForceActive	Active when bolt is controlled independently of striking plate (door)
0x0100	BoltLocked	Active when bolt is fully extended
0x0200	BoltUnlocked	Active when bolt is fully retracted
0x0400	BoltMovingOut	Active when bolt extends straight
0x0800	BoltMovingIn	Active when bolt is retracting
0x1000	Reserved	-
0x2000	Reserved	-
0x4000	Reserved	-
0x8000	Reserved	-

Table 5: Contents of Holding Register 1000 (Status Word)

5 Communication protocols

In order to be able to select the desired communication protocol, the "BUS-SELECT" input (pin 6 / yellow line) is provided. This is scanned during the switch-on process and the bus driver to be used is activated accordingly. To change the communication protocol, it is therefore necessary to restart the HD-Lock. **Attention! It is imperative to ensure that the selected communication protocol matches that of the network to which the HD-Lock is connected. If the wrong protocol is selected, the HD-Lock and/or the network may be damaged.**

If the "BUS-SELECT" input is left open or connected to "Ground/GND", the Modbus is selected.
When the "BUS-SELECT" input is connected to the supply voltage ("VCC"), the CAN bus is selected.

5.1 Modbus

The following factory settings apply to the Modbus:
- Transmission parameters
  - Transmission rate: 19200 baud
  - Parity: Even
  - Stop bits: 1
  - Byte length: 8
- Valid slave IDs can be set from 1-247 according to the Modbus standard, factory setting is ID 1. The HD-Lock also supports the address space from 248 - 254.

5.1.1 Modbus telegrams

The Modbus telegram consists of slave address, function code, data and CRC check. There must be a pause of at least 3.5 characters between these telegrams. The individual characters must not be more than 1.5 characters apart.

Slave address (ADDR)	Function Code (FC)	Data (DATA)	Checksum (CRC)
8 bit	8 bit	n * 8 bit	16 bit

Table 6: Modbus telegram

Slave address

The communication participants consist of a master, which has no address, and a number of slaves. With the Modbus standard, the slave addresses can be selected between address 1 and 247. In the factory, the units are marked with the slave address 1. However, the HD-Lock also supports the address range from 248 to 254. Addresses 0 and 255 are reserved as broadcast addresses.

Function code

Function codes indicate the purpose of the data transmission, e.g. whether bits or registers are to be read or written. These can be found under chapter 5.1.2 can be found.

Data

The information to be transmitted is received here. This field is divided into registers, number of registers to be transmitted and information to be read or saved

CRC check

The CRC checksum serves as a check word and is calculated over all bytes of the telegram. This is done by the sender and receiver to be able to detect transmission errors.

5.1.2 Function codes

In order to access the registers, various commands ("function codes") are used. These determine which register type is addressed, how many addresses are affected and whether the data is to be read or written. The following function codes are supported for this data area:

Function code	***Function***
FC03 "0x03"	**"read Holding Registers":** Reads one or more registers of the holding register group. Requires an address from which to read and the set of registers affected.
FC04 "0x04"	**"read input registers":** Reads one or more registers of the input register group. Requires an address from which to read and the set of registers involved.
FC06 "0x06"	**"write single holding register":** Describes a register of the holding register group. Requires the address of the register and the data to be written.
FC16 "0x1F"	**"write multiple holding** registers": Writes to one or more registers of the holding register group. Requires the start address from which to write, the number of registers to be written and the data to be written.
FC23 "0x8F"	**"read/write multiple holding** registers" Writes and reads multiple registers of the holding register group. Requires the start address from which to write and the number of registers to be written. Furthermore, the start address of the registers to be read and their number are required.

Table 7: Description of the "Function Codes

5.1.3 Broadcast telegrams

There are 2 broadcast addresses (0 and 255). These differ in the behaviour of the HDLocks -that receive them.

Address 0: An HD-Lock that receives such a broadcast telegram processes it, but will not reply.

Address 255: An HD-Lock that receives such a broadcast telegram processes it and returns an answer. Attention, this can lead to collisions on the bus.

As a result, address 0 is suitable for parameterising several HD-Locks. If, however, only one HD-Lock is connected and the address is not known, it can be read out via address 255 in order to find out the address.

5.2 CAN bus

The following factory settings apply to the CAN bus:

- Transmission parameters
  - Transmission rate: 250 kBaud

5.2.1 CAN bus messages

The CAN bus telegram is composed of arbitration field, control field, data field, checksum field and confirmation field.

Start Of Frame (SOF)	CAN base identifier (CAN ID)	Remote Transmission Request (RTR)	Identifier Extension (IDE)	Reserved (r0)	Data length code (DLC)	Datafield (DATA)	Checksum (CRC)	Acknowledge (ACK)	End Of Frame (EOF)	Intermission Frame Space (IFS)
1 bit	11 Bit	1 bit	1 bit	1 bit	4 bit	[1...8] x 8 bit	16 bit	2 Bit	7 bit	Min. 3 bit

Table 8: CAN 2.0A Bus Frame

Start Of Frame (SOF)	CAN Extended Identifier (CAN ID)	Substitute Remote Request (SRR)	Identifier Extension (IDE)	CAN Extended Identifier (CAN ID)	Remote Transmission Request (RTR)	Reserved (r1)	Reserved (r0)	Data length code (DLC)	Datafield (DATA)	Checksum (CRC)	Acknowledge (ACK)	End Of Frame (EOF)	Intermission Frame Space (IFS)
1 bit	11 Bit	1 bit	1 bit	18 Bit	1 bit	1 bit	1 bit	4 bit	[1...8] x 8 bit	16 bit	2 Bit	7 Bit	Min. 3 bit

Table 9: CAN 2.0B Bus Frame

Start Of Frame (SOF)

This (start) bit signals the beginning of a message by logical 0 (dominant bit). The falling edge from logic 1 to logic 0 is used to synchronise the bus participants.

CAN-Base/Extended-Identifier (CAN-ID)

The CAN base identifier (11 bits) or the CAN extended identifier (29 bits) identify the subsequent message and determine its priority. A low CAN ID has a higher priority. It should be noted that each CAN ID may only be used by one bus node at a time. With the CAN-Base-Identifier 4096 and with the CAN-Extended-Identifier 1'073'741'824 different messages can be distinguished. Important: Only the message is identified. The recipient is not specified. All messages are therefore basically broadcast messages.

Remote Transmission Request (RTR)

The RTR bit can be used to distinguish between two types of messages. If the bit is logical 0 (dominant), it is a message that contains data. If the bit is logical 1 (recessive), it is a request. In this case, a message is requested that has the same CAN ID. Example: In a bus with a control unit and a sensor, the current measured value is to be requested from the sensor. This would be transmitted in a data message with the CAN ID 42. The control unit now sends a request (RTR = 1) with the CAN ID 42. The sensor will then send the data message.

It should be noted that a request does not contain any data. The "Datafield" area is therefore omitted. However, the value for "Data Length Code" must be at least 1.

Identifier Extension (IDE)

This bit indicates whether the simple CAN base identifier (CAN 2.0A) or the CAN extended identifier (CAN 2.0B) is used. With logical 0 (dominant)→ CAN-Base-Identifier. For logical 1 (recessive)→ CANExtendedIdentifier-.

Reserved (r0, r1)

These bits are reserved for future developments.

Data Length Code (DLC)

These 4 bits determine the length of the "datafield". It should be noted that only values up to 8 are supported.

Datafield (DATA)

User data can be transmitted in this area, which is up to 8 bytes long.

Checksum (CRC)

The checksum is used for error detection by the recipient of the message. The cyclic redundancy check (CRC) is used for this. The last bit must always be set to logical 1 and serves as a separator.

Acknowledge (ACK)

The acknowledge of the CAN bus consists of two bits. The first of the two (acknowledge slot bit) contains the information about the correct reception. In the case of correct reception, the receiver responds logically 0 (dominant). It should be noted that every receiver sends a dominant level if it has received the message correctly. The bit alone therefore only provides information as to whether a receiver has received the message at all.

The second bit is the separator to the "End of Frame" field. This bit must be logic 1 (recessive).

End Of Frame (EOF)

The end of a message is signalled with 7 recessive bits.

Intermission Frame Space (IFS)

At least 3 recessive bits must be placed between two messages.

5.3 CAN messages from and to the HD-Lock

A "device ID" is used for communication with the HD-Lock. On the one hand, this is used so that an individual HD-Lock can be addressed specifically without occupying further CAN IDs, and on the other hand, it represents the CAN ID with which the HD-Lock provides all responses. This "device ID" can be defined as a simple CAN base identifier or as an extended identifier.

In the delivery state, the "unit ID" is described with the serial number as the extended identifier.

5.3.1 CAN messages to the HD Lock

The HD-Lock accepts the following CAN IDs and processes them. For this purpose, the "target address" (byte 0-3) is compared with the respective own device ID. If these two values match, the rest of the message is processed as described below.

Message	CAN ID	Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7	Length
Command	0x700	Destination address (0→ Broadcast)	Lo Cmd	Hi Cmd	Lo param	Hi Param	8
SetID	0x701	Destination address (0→ Broadcast)	Lo newID	Hi newID	UpLo newID	UpHi newID	8
Read input register	0x702	Destination address (0→ Broadcast)	Lo Addr	Hi Addr	Lo Count	Hi Count	8
Read holding register	0x703	Destination address (0→ Broadcast)	Lo Addr	Hi Addr	Lo Count	Hi Count	8
Write holding register	0x704	Destination address (0→ Broadcast)	Lo Addr	Hi Addr	Lo Count	Hi Count	8
Silent write holding register	0x705	Destination address (0→ Broadcast)	Lo Addr	Hi Addr	Lo Count	Hi Count	8
FWU erase sector	0x706	Destination address (0→ Broadcast)	Lo Addr	Hi Addr			6
FWU write flash	0x707	Destination address (0→ Broadcast)	Sector offset	Sector offset			8

Table 10: Accepted CAN IDs

Command

Various commands can be sent to one or all HD-Locks on the bus via these messages. If the destination address matches the device address or is 0 (broadcast), bytes 4-7 are processed. Bytes 4-5 determine which command from the following list will be executed.

Command	Designation	Parameter
0	Set lock request source	New value
1	Set CAN baudrate	New value
2	Enter firmware update mode	-
3	Enter bootloader update mode	-
4	Leave update mode	-

Table 11: CAN commands

- **"Set lock request source" command**: Sets the condition that must be fulfilled for the bolt to extend. The following conditions are available (Attention! Only the LSB byte is permanently stored. The MSB byte is reset to 0 after a reset):

Value	Symbol	Meaning	Dependence
0x00	IrsNever	Unlock	Door closed
0x01	IrsAlways	Lock	Door closed
0x02	IrsOverrange	Lock when Hall sensors are overridden (address 8 of input register 1000)	Door closed
0x03	IrsLine	Latched when input inactive (address 0 of input register 1000)	Door closed
0x04	IrsNotLine	Locked when input active (address 0 of input register 1000)	Door closed
0x05	IrsToggleOnLineEdge	Locked when an edge is detected at the input (address 0 of input register 1000)	Door closed
0x06	IrsToggleOnLinePEdge	Locked when a positive edge is detected at the input (address 0 of input register 1000)	Door closed
0x07	IrsToggleOnLineNEdge	Locked when a negative edge is detected at the input (address 0 of input register 1000)	Door closed
0x80	IrsNever	Unlock	None
0x81	IrsAlways	Lock	None
0x82	IrsOverrange	Lock when Hall sensors are overridden (address 8 of input register 1000 active)	None
0x83	IrsLine	Locked if input inactive (address 0 of input register 1000)	None
0x84	IrsNotLine	Locked when input is active (address 0 of input register 1000)	None
0x85	IrsToggleOnLineEdge	Locked when an edge is detected at the input (address 0 of input register 1000)	None
0x86	IrsToggleOnLinePEdge	Locked when a positive edge is detected at the input (address 0 of input register 1000)	None
0x87	IrsToggleOnLineNEdge	Locked when a negative edge is detected at the input (address 0 of input register 1000)	None

Table 12: "Lock Request" Conditions

- **"Set CAN baudrate" command:** Sets the baud rate to be used by the CAN bus. This cannot be chosen arbitrarily, but must be selected from the following list:

Value	Baud rate	Bus length max.
0	5 Kbit/s	10'000 m
1	10 Kbit/s	5'000 m
2	20 Kbit/s	2'500 m
3	33.333 Kbit/s	"
4	50 Kbit/s	1'000 m
5	83.333 Kbit/s	"
6	100 Kbit/s	"
7	125 Kbit/s	500 m
8	250 Kbit/s	250 m
9	500 Kbit/s	100 m
10	800 Kbit/s	"
11	1 Mbit/s	25 m

Table 13: CAN bus baud rates

- **"Enter firmware update mode" command, "Enter bootloader update mode" command and "Leave update mode" command:**
  These commands are used for firmware updates and should not be used by the user.

SetID

The SetID message can be used to change the unit ID. Since the unit ID is described in the factory with the serial number as an extended identifier, it is usually necessary to make a change. For identifiers greater than 0x7FF, an extended identifier is automatically set, until then a simple CAN base identifier. If an extended identifier up to 0x000007FF is to be used, the MSB must be set (newID = newID | 0x80000000). If an identifier is set equal to 0, the serial number is again used as the device ID.

Read input Register

This message allows the contents of one or more input registers to be queried. The HD-Lock whose device ID matches the destination address responds to this message by sending back the contents of the requested registers (see section 5.3.2).

Read holding register

Identical to the "Read input register" message except that holding registers are queried.

Write holding register

This message allows the contents of one or more holding registers to be written to. The HD-Lock whose device ID matches the destination address responds to this message by writing to the selected holding registers and then sending the contents back (see section 5.3.2). Please note that holding registers that are write-protected by the firmware cannot be changed. In this case, the HD-Lock reacts identically to the "Read holding register" message.

Silent write holding register

Identical to the "Write holding register" message except that no response is given.

FWU erase sector, FWU write flash

These messages are for firmware updates and should not be used by users themselves.

5.3.2 CAN messages from the HD-Lock

The HD-Lock always sends messages under the device ID. The content of the messages is identified via the first two data bytes. The possible contents are listed below:

Message	CAN ID	Byte 0	Byte 1	Byte 2-7 (data are unsigned)	Length	Time of transmission
Status	Device ID	0	0	2 bytes each: [Status], [motorPos], [(proximity<< 6)+LockRequestSource]	8	Every 1000 ms
FWU status	Device ID	1	0	2 bytes each: [flashing 1/0], [Application/BootLoader 1/0]	6	Every 1000 ms
Input register	Device ID	0	1	1 byte each: [LoAddr], [HiAddr], [LoValue], [HiValue]	6	Response to read input register
Holding register	Device ID	1	1	1 byte each: [LoAddr], [HiAddr], [LoValue], [HiValue]	6	Response to read/write holding register

Status

Each HD-Lock periodically reports the current status as long as it is not in firmware update mode. A message is sent with the CAN ID equal to the device ID whose first two data bytes are equal to 0. This status message then contains the following information:

- Status: The status word [Holding Register 1000] (see section 4.3).
- MotorPos: Current count of the motor increment counter. (close to 0 = bolt retracted, close 130 = bolt extended)
- Proximity: Calculated position of the striking plate via HD-Lock (0 = congruent, 100 detection range left)
- LockRequestSource: Condition currently used by the HD-Lock to extend the bolt [Holding Register 4].

FWU status

While the HD-Lock is in firmware update mode, the current status of the update is reported periodically. A message is sent with the CAN ID equal to the unit ID whose first data byte is equal to 1 and the second to 0. This status message then contains the following information:

- Flashing: 0→ currently not being flashed, 1→ currently being flashed
- Application/Boot Loader: 0→ Boot Loader is being updated, 1→ Firmware is being updated

Input register

In response to the "Read input register" message, the HD-Lock sends a message. In doing so, a message is sent with the CAN ID equal to the device ID whose first data byte is equal to 0 and the second is equal to 1. This response then contains the following information:

- LoAddr: Lower part of the register address that was read.
- HiAddr: Upper part of the register address that was read.
- LoValue: Lower part of the register content.
- HiValue: Upper part of the register content.

Holding register

In response to the "read holding register" and the "write holding register" message, the HD-Lock sends a message. A message is sent with the CAN ID equal to the device ID whose first data byte is equal to 1. This response then contains the following information:

- LoAddr: Lower part of the register address that was read/written.
- HiAddr: Upper part of the register address that was read/written.
- LoValue: Lower part of the register content.
- HiValue: Upper part of the register content.